Mitigating Poisoning Attacks on Machine Learning Models: A Data Provenance Based Approach

Nathalie Baracaldo
IBM Research
San Jose, California
baracald@us.ibm.com

Bryant Chen
IBM Research
San Jose, California
bryant.chen@ibm.com

Heiko Ludwig
IBM Research
San Jose, California
hludwig@us.ibm.com

Jaehoon Amir Safavi
IBM Research
San Jose, California
amir.safavi@ibm.com

ABSTRACT

The use of machine learning models has become ubiquitous. Their predictions are used to make decisions about healthcare, security, investments and many other critical applications. Given this pervasiveness, it is not surprising that adversaries have an incentive to manipulate machine learning models to their advantage. One way of manipulating a model is through a *poisoning* or *causative* attack in which the adversary feeds carefully crafted poisonous data points into the training set. Taking advantage of recently developed tamper-free provenance frameworks, we present a methodology that uses contextual information about the origin and transformation of data points in the training set to identify poisonous data, thereby enabling online and regularly re-trained machine learning applications to consume data sources in potentially adversarial environments. To the best of our knowledge, this is the first approach to incorporate provenance information as part of a filtering algorithm to detect causative attacks. We present two variations of the methodology–one tailored to partially trusted data sets and the other to fully untrusted data sets. Finally, we evaluate our methodology against existing methods to detect poison data and show an improvement in the detection rate.

CCS CONCEPTS

• **Security and privacy → Software and application security**; Use https://dl.acm.org/ccs.cfm to generate actual concepts section for your paper; • **Computing methodologies → Machine learning**; **Supervised learning**;

KEYWORDS

Adversarial machine learning; supervised learning; Internet of the Things; IoT; provenance; security; poisoning attacks; causative attacks

1 INTRODUCTION

The reliance of machine learning methods on quality training data presents a security vulnerability in which adversaries may inject poisonous samples into the training dataset to manipulate the learned classifier. A highly-publicized example of this is the recent attack on Microsoft's AI chat bot, Tay, which learned offensive and racist language from Twitter users. Defending against these types of attacks, called poisoning or causative attacks, is particularly challenging in online learning and other environments where the model must be periodically retrained to account for dataset shifts.

Existing approaches to identify poison data points focus on analyzing only the data received for training, and a survey can be found in [3]. In some cases, however, there exists contextual information that can guide the detection of poisonous data points. Provenance data refers to the lineage or meta-data associated with a data point and shows the operations that led to its creation, origin and manipulation. This may include information about the device from which the data was gathered, its firmware version, user id, and timestamp among others.

Several tamper-free provenance frameworks have been proposed in the literature to collect lineage information while preventing history re-writing, repudiation and fabrication of provenance data [1, 2, 8, 10, 11, 17, 18]. These frameworks ensure that provenance records are cryptographically protected using methodologies that incorporate technologies such as blockchain [1], physical unclonable functions [2], trusted computing platforms [11] and others.

However, provenance frameworks are mainly used for accountability and forensic purposes. Hence, provenance information is only used after a poisoned machine learning model has been deployed, the damage has been done, and the attack detected. In contrast, we propose a proactive methodology to use provenance data to detect poisonous data.

To the best of our knowledge, our method is the first defense strategy that makes use of data provenance to filter untrusted data points and prevent poisoning attacks. We use provenance meta-data to segment the untrusted data into groups where the probability of poisoning is highly correlated across samples in each group. For example, in an IoT environment, an adversary is likely only able to compromise a portion of the data-collecting sensors. Using data provenance, we can segment the untrusted dataset by sensor

AISec'17, November 3, 2017, Dallas, TX, USA

ACM ISBN 978-1-4503-5202-4/17/11…$15.00
https://doi.org/10.1145/3128572.3140450

and evaluate the data in each segment together. Alternatively, the dataset could be segmented by firmware version, location, user account, or other contextual information that is indicative of the poisoning process. Once the training data has been segmented appropriately, data points in each segment are evaluated together by comparing the performance of the classifier trained with and without that group.

We present two variations of our methodology depending on the type of data available: *partially trusted* and *fully untrusted*. The case of partially trusted data sets arises when it is possible to verify some of the data collected or when some data sources can be trusted because they are in a highly controlled setting. The case of fully untrusted data occurs when it is impractical, too costly, or infeasible to verify a portion of the data set.

Two prior methods, Reject on Negative Impact (RONI) [13] and the Probability of Sufficiency (PS) method by [7], take a similar approach of detecting poison data by evaluating the effect of individual data points on the performance of the trained model. Both of these methods evaluate the model by comparing its performance on a trusted data set. When a trusted data set is available, our method also evaluates performance in this manner. However, by evaluating the data in each segment together, our method makes the effect of poisonous data more obvious, enabling higher detection rates. Additionally, the detection process is more scalable because it reduces the number of times the model needs to be retrained to a fraction of the total number of untrusted points. Finally, we will show that data provenance allows this general approach to be robustly applied to environments where no trusted data is available for evaluation purposes. Niether RONI nor PS can be applied in such environments.

The **contributions** of this paper are the following:

- We propose a novel method for detecting and filtering poisonous data collected to train an arbitrary supervised learning model. In particular, this method uses data provenance to identify groups of data whose likelihood of being poisoned are highly correlated.
- We present two flavors of our provenance-based defense for cases when *partially trusted* and *fully untrusted* datasets are available. Both cases cover a wide range of applications.
- We evaluate our method to detect poison data generated by two methods of [6] and [19]. We find that using our defense as a filter prior to training significantly improves classification performance of models trained on both partially trusted data sets. Our results show that our method outperforms RONI.

The rest of this paper is organized as follows. In Section 2 we present in detail the threat model. Then, in Section 3 we introduce our provenance defense to identify poisonous data when a partially trusted data set is available. In Section 4, we present a second methodology to deal with fully untrusted data. We evaluate our approach in Section 5. We present the related work in Section 6 and conclusions in Section 7.

2 THREAT MODEL AND TERMINOLOGY

In this paper, we consider an adversary that aims to reduce the overall accuracy of the machine learning model to render it unusable.

We assume the existence of a provenance framework deployed to record the lineage of data points received for training. This provenance framework protects the integrity of provenance data and ensures non-repudiation and non-fabrication of information sent to the training system. The provenance framework provides a *provenance record* for each data point collected that contains one or more *provenance features* reflecting its lineage. A value for a provenance feature, e.g., a specific video camera, a Twitter account, or a specific firmware version., is called a *provenance signature*. The set of collected data points sharing a provenance signature is called the *data segment* of this signature. As we highlighted in the introduction, multiple implementations exist to ensure that provenance records cannot be forged and remain immutable.

In other cases, an explicit provenance framework may not be in place, but we can nevertheless consider certain features to be trusted and indicative of the origin and lineage of the data. For example, if the training data consists of tweets, then the originating Twitter account can be considered as a provenance feature for the purpose of our method. While an account might be hacked, the account from which a particular tweet originated can generally be considered to be accurate. Similarly, an adversary that attempts to manipulate a classifier trained to identify fraudulent credit card transactions may poison the training data by misreporting transactions to the credit card company. In this case, the adversary can manipulate various aspects of the transaction and its classification but cannot manipulate the account to which the transaction is posted. Twitter and credit card accounts are also examples of features that are indicative of how poisonous points might be clustered, as adversaries are likely only able to manipulate a small portion of them.

We allow the adversary to observe or acquire data that is similar to the one used to train the algorithm and can, therefore, use this information to craft poisonous data points. We also allow the adversary to modify the features extracted from data points and their labels when crafting poisonous data. In real systems, an adversary can typically only compromise a sub-set of data sources - compromising all of them can be expensive in terms of time and resources spent by the adversary. Hence, we assume that the adversary cannot compromise all data sources that send data to the training system. That is, the adversary can modify data points sharing certain provenance signatures. In Section 5, we will evaluate the effects of violating this assumption.

3 PROVENANCE DEFENSE FOR PARTIALLY TRUSTED DATA

In this section, we present our provenance-based poisoning defense method for environments where the collected data is partially trusted. By *partially trusted*, we mean that some of the data points in the collected data are assumed to be legitimate (not poisoned). In real-world scenarios, obtaining partially trusted training data can be achieved through manual curation of the collected data or through trusted sources of data.

Our method is agnostic to the supervised machine learning algorithm used, and, in theory, could also be applied to unsupervised algorithms. However, we restrict our analysis to supervised learning algorithms so that the performance of the trained models can be more easily compared and evaluated.

This method takes as input

(1) a supervised machine learning algorithm,
(2) a partially trusted training data set collected for the purposes of training the machine learning classifier, which consists of two parts–a trusted set and an untrusted data set,
(3) a secure and trusted provenance data set which consists of meta-data describing the origin and lineage of each data point in the untrusted portion of the training set,
(4) and a provenance feature in the provenance data set that is indicative of how poisonous points will be clustered in the untrusted portion of the data set.

Algorithm 1 findPoisonDataPartiallyTrusted(D, D_T, $\mathcal{F}$)

Input: D := all data points, D_T := trusted data points (trusted set), $\mathcal{F}$:= Provenance feature to be used for segmentation

Output: Set of data points that are suspected of being poisonous.

```
$D_{poisoned} \leftarrow \emptyset$
$D_U \leftarrow D \setminus D_T$ {Untrusted data}
$F \leftarrow$ segmentByProvenanceFeature($D_U$, $\mathcal{F}$)
for all $\langle D_i, Sig_i \rangle \in F$ do
    $Model_{filtered} \leftarrow$ trainModel($D_U \setminus D_i$)
    $Model_{unfiltered} \leftarrow$ trainModel($D_U$)
    if performance($Model_{unfiltered}$, $D_T$) < performance($Model_{filtered}$, $D_T$) ) then
        $D_{poisoned} \leftarrow D_{poisoned} \cup \langle D_i, Sig_i \rangle$ {Flag as suspicious}
        $D_U \leftarrow D_U \setminus D_i$ {Remove from training set}
    end if
end for
return $D_{poisoned}$
```

Given the above inputs, our method follows the process described in Algorithm 1 and depicted in Figure 1. First, each data point in the untrusted training data set is linked with its own provenance record, which is shown in the green table in Figure 1. Provenance records may have multiple features, such as device identification number, firmware version of the device, timestamp and others. Then, the untrusted data is segmented so that each segment shares the same provenance signature. In the credit card fraud example, the data set consisting of credit card transactions is segmented by the account to which the transaction is posted. Each segment is then evaluated for poison by using the machine learning algorithm to train classifiers with and without that segment of data. If the classifier trained without the segment (filtered model) performs better than the one trained with it (unfiltered model) on the trusted test set, then we consider that segment to be poisoned and remove it from the untrusted data set. In our experiments, we measure performance using the classification accuracy, but, in theory, any performance metric can be used.

4 PROVENANCE DEFENSE FOR FULLY UNTRUSTED DATA SETS

In some scenarios, it is difficult or even infeasible to obtain a partially trusted data set due to cost associated with manual data verification, such as paying annotators to verify labels, and real-time requirements that preclude data verification. To address these scenarios, we present a provenance based poison detection mechanism that works even if *all* data collected for re-training is untrusted.

To apply our method to fully untrusted data sets, we propose the following procedure.

(1) Segment the data by signature according to the selected provenance feature.
(2) Split the data set randomly into a training portion and an evaluation portion.
(3) For each signature in the selected provenance feature:
    (a) train two models–one with all of the training data and one with the corresponding segment in the training data removed;
    (b) evaluate both models on the evaluation set with the corresponding segment removed;
    (c) permanently remove the segments from both the training and evaluation set if the model trained without it performed better.

Algorithm 2 findPoisonDataFullyUntrusted(D_U, $\mathcal{F}$)

Input: D_U := all data points (all are untrusted), $\mathcal{F}$:= Provenance feature to be used for segmentation

Output: Set of data points that are suspected of being poisonous.

```
$D_{poisoned} \leftarrow \emptyset$
$F \leftarrow$ segmentByProvenanceFeature($D_U$, $\mathcal{F}$)
$F_{train} \leftarrow \emptyset$, $F_{eval} \leftarrow \emptyset$
for all $\langle D_i, Sig_i \rangle \in F$ do
    Randomly assign half of the data in $D_i$ to $F_{train}$ and half to $F_{eval}$
end for
for all $\langle D_i, Sig_i \rangle \in F_{train}$ do
    $Model_{filtered} \leftarrow$ trainModel($D_{train} \setminus D_i$)
    $Model_{unfiltered} \leftarrow$ trainModel($D_{train}$)
    $\langle D_{eval_i}, Sig_i \rangle \leftarrow$ getSegment($F_{eval}$, $Sig_i$)
    $D_{filteredEval} \leftarrow D_{eval} \setminus D_{eval_i}$
    if performance($Model_{unfiltered}$, $D_{filteredEval}$) < performance($Model_{filtered}$, $D_{filteredEval}$) ) then
        $D_{poisoned} \leftarrow D_{poisoned} \cup \langle D_i, Sig_i \rangle$ {Flag as suspecious}
        $D_{train} \leftarrow D_{train} \setminus D_i$ {Remove from training set}
        $D_{eval} \leftarrow D_{eval} \setminus D_{filteredEval}$ {Remove from validation set}
    end if
end for
return $D_{poisoned}$
```

This method is described more formally by Algorithm 2. By removing the corresponding points from the evaluation set when determining whether a particular segment is compromised, we prevent the data source from manipulating its own evaluation. Otherwise, an adversary that has managed to compromise a particular

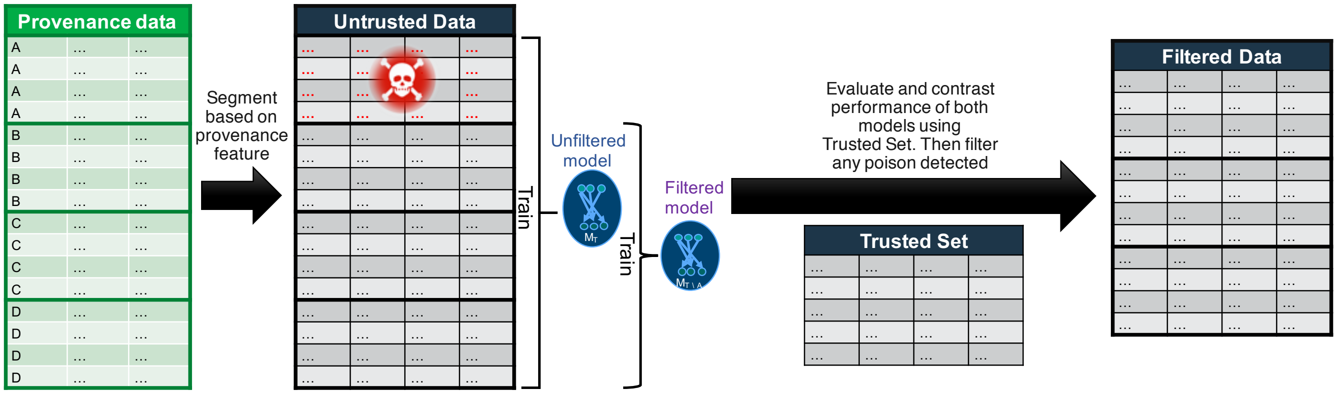

Provenance data
A
B
C
D
Segment based on provenance feature
Untrusted Data
Train
Unfiltered model
M_T
Train
Filtered model
M_T\A
Evaluate and contrast performance of both models using Trusted Set. Then filter any poison detected
Trusted Set
Filtered Data

Figure 1: Overview of our provenance defense for partially trusted data. Each data point in the untrusted set is associated with a provenance record consisting of one or more provenance features. Algorithm 1 presents the detailed procedure.

device can use it to not only poison the machine learning classifier, but also interfere with the evaluation process, allowing poisonous points to evade detection.

To illustrate, we performed a simple logistic regression simulation using the following setup. First, 200 "legitimate" data points, $\{x_i, y_i\}$ were generated in the following way. $\{x_i\}$ was sampled from a normal distribution with mean 0 and variance 10, and the y_i sampled from a distribution where $P(y_i = 1|x_i) = \frac{1}{1+e^{-x_i}}$. Next, we inserted 10 poison data points with $x = 10$ and $y = 0$ and another 20 poison data points with $x = 1$ and $y = 0$. Finally, we randomly selected half of the total 240 points to be the training set and half to be the evaluation set.

When evaluating the compromised source, we will not be able to detect that it is compromised without removing that source's data from the evaluation set. This is illustrated in Figure 2. Figure 2a shows the performance of the logistic regression model trained on the entire dataset, and Figure 2b shows the performance of the logistic regression model with all of the poison data removed. Here we see that removing the poison data shifts the decision boundary from 2.84 to -0.20, much closer to the true boundary at 0. However, the poison data at $x = 1$ in the validation set shifts from being classified correctly to being classified incorrectly so that removing the poison data actually decreased the accuracy from 82% to 78%. As a result, the poisonous data is not detected. In contrast, if the data from the source in question is also removed from the evaluation set, then the accuracy increases 82% to 93% (see Figure 2c), and the source is correctly identified as compromised.

The above example highlights the key role that data provenance plays in fully untrusted environments. Without data provenance, there is no way to link the data in the training set to the data in the evaluation set. As a result, it is not clear how to remove the influence of poisonous data in the evaluation process.

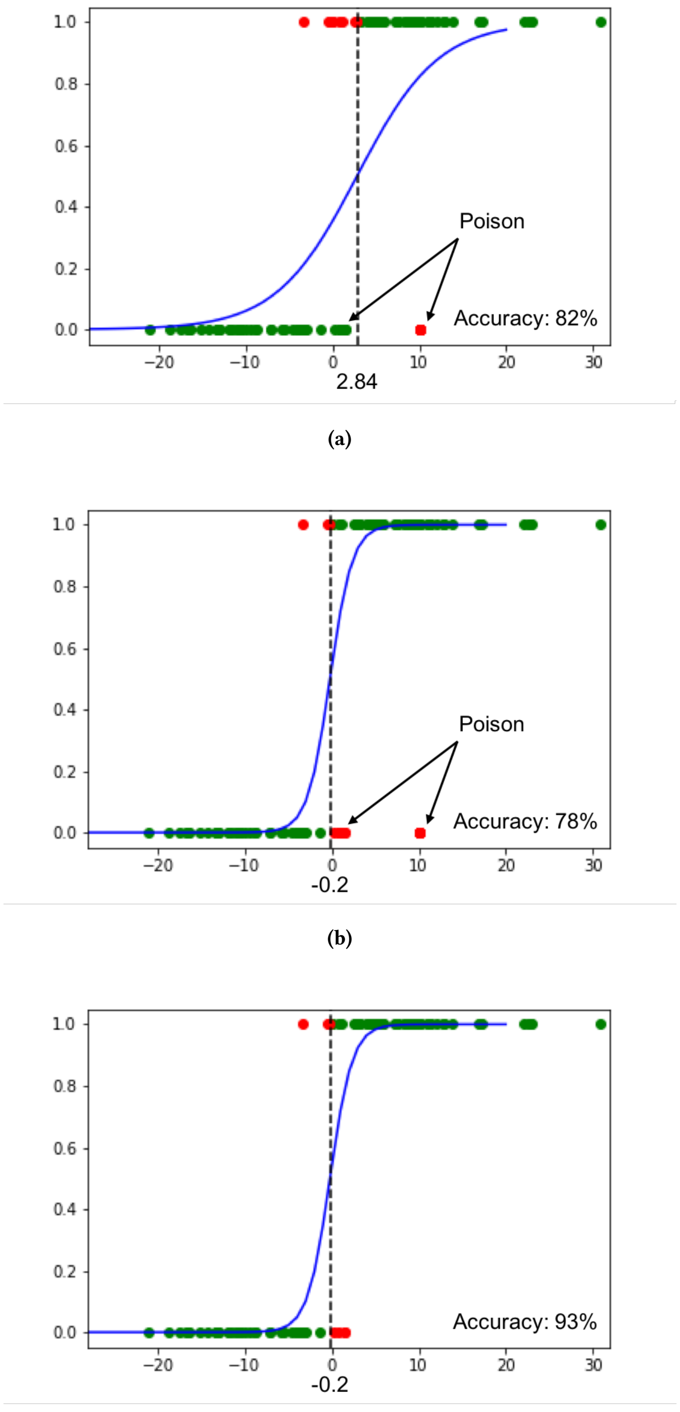
Lastly, our methods for partially trusted and fully untrusted data sets require that the model is retrained k times, where k is the number of data segments (e.g. number of credit card accounts). In contrast, prior methods like RONI require that the model be trained at least once for each data point. Since each data segment generally contains many data points, our method requires that the model be retrained a fraction of the number of times that RONI requires.

5 EXPERIMENTAL EVALUATION

To evaluate our approach, we generated an IoT scenario where multiple devices contribute data points used to train a model. To ensure the evaluation is fair, we evaluated our defense under two different poisoning mechanisms previously proposed in the literature [6, 19]. Both mechanisms target support vector machines (SVMs). For each type of poison, we used the following procedure. We first generated legitimate data points and poison data points using the corresponding poisoning methodology [6, 19] and defined the number of devices in the system. To generate provenance data for each data point, each IoT device was assigned the same number of contributing data points. To facilitate the analysis, we evaluated our approach under a fully dishonest and honest assumption, where compromised devices contribute only poisonous data points and honest devices contribute only legitimate data points.

We compared our approach with two defenses RONI [13] and RONI with calibration [15]. Given that [15] outperformed [13], we only report the results for Calibrated RONI and use it as baseline. Both the provenance defense and Calibrated RONI use a trusted set. Calibrated RONI requires the following inputs as parameters: number of data points used for calibration, validation, baseline and sampling repetitions, which were set to 50, 100, 20 and 10, respectively. We refer the reader to [15] for a detailed explanation on how these parameters are used by the baseline. To compare both methodologies, we used the same size for the trusted set. In accordance with [15], the trusted set used for Calibrated RONI is split into the calibration, validation and baseline.

Finally, we separated an independent test set of 5000 legitimate data points uniquely used for benchmarking purposes. With this benchmarking data set, we assessed the accuracy of four models: *Perfect detection* model trained using only legitimate data points, *no-defense* model trained using all data points received by the system, *provenance defense* model trained after filtering data points classified as poisonous by our defense and *baseline defense* model trained after filtering data points identified as poison by the calibrated RONI.



(a)

(b)

(c)

Figure 2: (a) Performance of logistic regression model trained on all data, including poison (b) Performance of logistic regression model trained with poison data removed from training set but not the evaluation set (c) Performance of logistic regression model trained and evaluated with poison data removed by our provenance method.

5.1 Effectiveness under poison I:

We assessed our defense against the poison attack presented in [19] using the same synthetic dataset used in their paper, which has two features and two classes. Their poison algorithm receives two parameters: i) *attack factor* that varies from 0 to 1, where one is the most aggressive attack and 0 makes no changes, and ii) *separation* that can be set to *small, medium* or *large*, which indicates the amount of separation between the two classes. The attack consists of selecting an attack class, taking a legitimate data point, then flipping the label and moving the feature set closer to the targeted class, according to the separation parameter. The following results were generated for an attack factor of 0.5 and a small separation; we present the results for these parameters because a small separation results in the most challenging poison detection case. The figures shown were generated by running 20 repetitions and averaging their results.

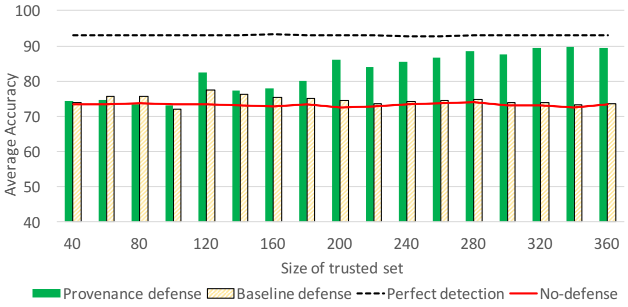


Figure 3: Effect of increasing the size of the trusted set on the average accuracy achieved under poison I

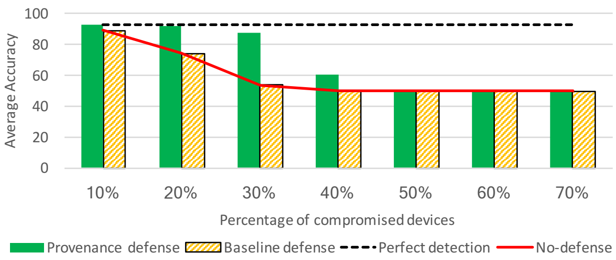


Figure 4: Effect of increasing the percentage of compromised devices on the average accuracy achieved under poison I.

Effect of trusted set size: Our first experiment assesses the effect of the size of the trusted set used. In this experiment, we set the number of total training points to 1000 and the number of poisonous data points to 200. The total number of honest and dishonest devices were kept to ten and two, respectively.

The results are presented in Figure 3, where the effectiveness of the evaluated poison is reflected by the distance between the perfect detection and the no-defense lines. Not using a poison detection mechanism would allow the adversary on average to reduce the accuracy of the system by 20%.

The bars in Figure 3 show the accuracy achieved by each defense as a function of the trusted size. Our experiments indicate that for poison I, at least 120 trusted data points are needed before the provenance method improves over *no defense*. By 280 data points, it has converged and the final accuracy achieved is close to *perfect detection*. In contrast, the baseline defense performs poorly when detecting this type of poison, even at larger trusted set sizes.

Effect of number of compromised devices: In our next experiment presented in Figure 4, we evaluated the effect of increasing the number of compromised devices while maintaining the total number of devices fixed. The total number of data points contributed by each device was kept to 100 in this experiment.

Although it seems unlikely for an adversary to compromise more than half of the total number of devices in real systems, we nevertheless include such scenarios for analysis purposes. In scenarios where 10% to 40% of devices compromised, Figure 4 shows that applying the provenance defense results in a model with higher accuracy than the baseline. When 50% or more of the devices are compromised, neither method is able to improve the accuracy of the model and the number of points detected as poisonous. In such environments, it is not surprising that the provenance defense performs poorly since both the full and partial model are trained on data that is largely poisonous.

For future work, it would be interesting to explore the following variant of the provenance defense method. First, the trusted portion of the data is split into a training set and an evaluation set. When evaluating a particular untrusted data source, a model is trained on the trusted training set and compared to a model trained on the trusted training set with data from the untrusted source added in. If the latter model performs worse, then we consider that portion to be poisoned. We suspect that such a method would perform better in environments where a large portion of the untrusted data is poisoned and an adequate amount of trusted data is available.

Effect of number of data points contributed per device: We repeated the previous experiment changing the number of data points contributed by each device and present the results in Figure 5. The table shows the average accuracy of perfect detection, no-defense and the two evaluated defenses. The last column presents the improvement achieved by the provenance defense with respect to the baseline. As was the case in the previous experiment, as number of devices compromised increases, the ability to detect poison by both defenses decreases.

The last column of the table shows the average improvement of our defense with respect to the baseline for different number of data points contributed by each device. We can see that the number of data points per device does have an impact on the average improvement. In particular, as the number of data points per device increases, our solution tends to have a smaller improvement over the benchmark.

This result suggests that it is necessary to carefully decide how many data points per provenance signature needs to be evaluated at the same time by our solution to avoid false positives that have a very large negative impact. An interesting research direction consists in evaluating different ways in which provenance data can be segmented to achieve this type of result. For example, it is possible to segment according to a multiple provenance features, where each segment carries the same provenance signature for multiple features.

5.2 Effectiveness under poison II:

We also assessed our defense under poisonous data generated by the method presented by Biggio et. al [6]. Their method uses gradient ascent to update a single attack point that will be added to an SVM. By making some small changes to the algorithm, we created multiple poisonous points. We use the same dataset presented in their paper, the MNIST dataset of handwritten digits. Our experiments classified digits 4 and 0. The poisoning algorithm requires as input the number of repetitions of the poisoning algorithm, which we kept to four.

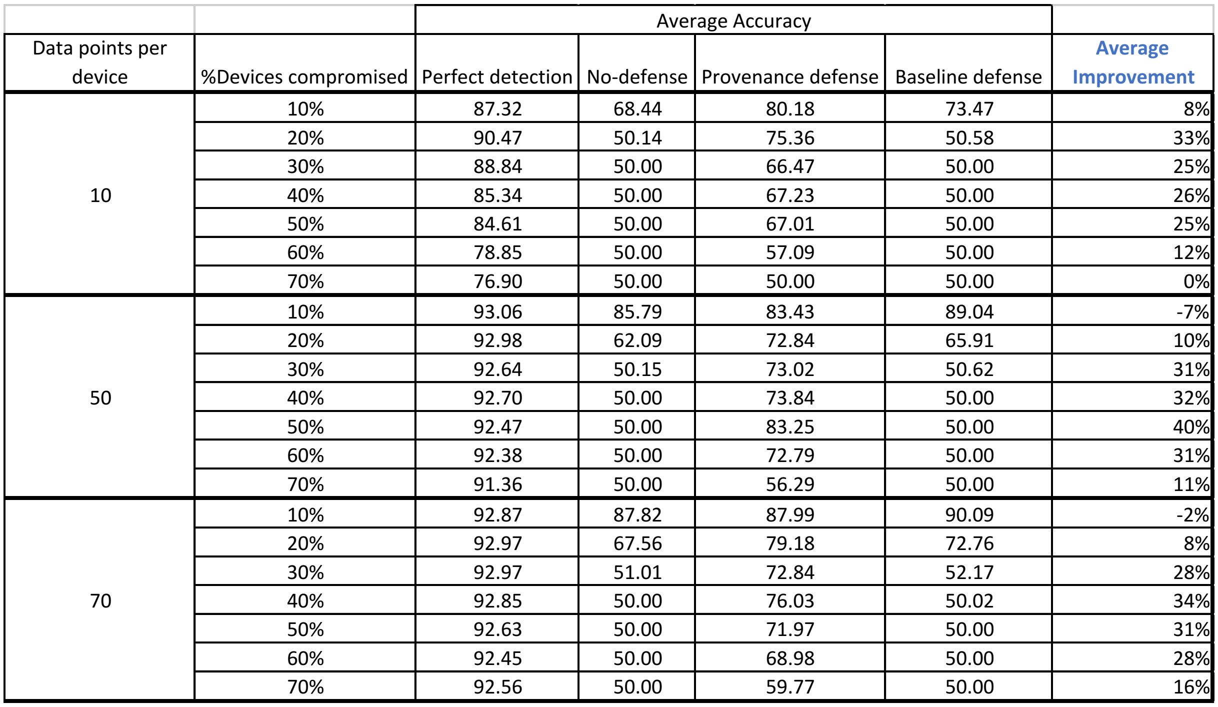
The figures for these experiments are the result of averaging five experiment trials. The total number of honest and dishonest devices were kept to ten and two, respectively. Unless explicitly mentioned, the total number of training data points was 120, the number of poison data points was set to 20 and the size of the trusted set was set to 120.

In Figure 6, we present the effect of increasing the trusted set size, keeping the rest of the parameters constant. The results are shown in Figures 6a and 6b. Even at 90 data points, the provenance defense greatly improves the performance of the final classifier. In contrast, at least 120 data points are needed before the baseline is able to improve over *no defense*. By 150 data points, the ability of the provenance method to improve the model accuracy has converged and performs nearly as well as *perfect detection*. While the baseline performs better for poison II than for poison I, it still performs worse than the provenance defense, regardless of the number of trusted points.

6 RELATED WORK

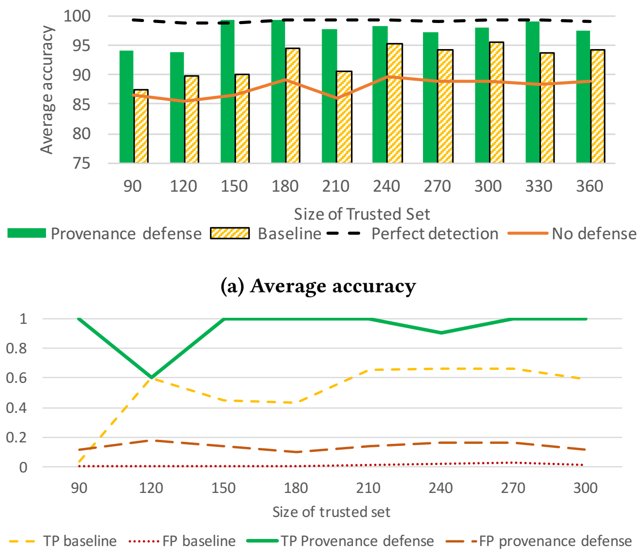
Many provenance frameworks have been proposed in the literature [1, 2, 8, 10, 17, 18] to ensure the lineage of data can be tracked for accountability purposes. These approaches focus on cryptographically preserving the history of data, non-fabrication and non-repudiation. However, to the best of our knowledge this is the first approach to use provenance information as an integral component to defend against poisoning attacks.

The use of machine learning systems in critical applications has drastically increased and with it the number of efforts to identify security vulnerabilities and defenses. Recent surveys in this area include [3, 4, 9, 16]. In this paper, we have focus on poisoning attacks, a.k.a. causative attacks, that target the training stage of the model. The closest related work is *Reject On Negative Impact* (RONI) methodology proposed by Nelson et al. in [13] and further enhanced in [15], where a calibration methodology to evaluate the performance of a model was included. These approaches assume the existence of a partially trusted data set. Our approach differs from these methodologies in that it makes use of provenance information that contains contextual cues to expedite the evaluation of untrusted data points. A detailed comparison of our approach and these two methodologies was presented in Section 5, where we



		Average Accuracy				
Data points per device	%Devices compromised	Perfect detection	No-defense	Provenance defense	Baseline defense	Average Improvement
10	10%	87.32	68.44	80.18	73.47	8%
	20%	90.47	50.14	75.36	50.58	33%
	30%	88.84	50.00	66.47	50.00	25%
	40%	85.34	50.00	67.23	50.00	26%
	50%	84.61	50.00	67.01	50.00	25%
	60%	78.85	50.00	57.09	50.00	12%
	70%	76.90	50.00	50.00	50.00	0%
50	10%	93.06	85.79	83.43	89.04	-7%
	20%	92.98	62.09	72.84	65.91	10%
	30%	92.64	50.15	73.02	50.62	31%
	40%	92.70	50.00	73.84	50.00	32%
	50%	92.47	50.00	83.25	50.00	40%
	60%	92.38	50.00	72.79	50.00	31%
	70%	91.36	50.00	56.29	50.00	11%
70	10%	92.87	87.82	87.99	90.09	-2%
	20%	92.97	67.56	79.18	72.76	8%
	30%	92.97	51.01	72.84	52.17	28%
	40%	92.85	50.00	76.03	50.02	34%
	50%	92.63	50.00	71.97	50.00	31%
	60%	92.45	50.00	68.98	50.00	28%
	70%	92.56	50.00	59.77	50.00	16%

Figure 5: Effect of the number of data points contributed per device on the accuracy and average improvement of the proposed method.



(a) Average accuracy

(b) Average true positive rate (i.e. recall) and false positive rate (i.e. fall-out)

Figure 6: Effect of increasing the size of the trusted set for poison II

showed the proposed provenance defense outperforms these two methodologies detecting poison.

Several approaches to poison models have been proposed in the literature. Zhou et al. [19] proposed two attack models for poisoning SVMs, as well as optimal SVM learning strategies against the proposed attack models. In contrast, the proposed provenance defense does not require a priori knowledge of the type of poison injected by adversaries. We also experimentally show that the proposed methodology is resilient against this type of poison. Biggio et al. [6] proposed an attack to SVMs where an adversary can manipulate all features of training data by running a gradient ascent optimization problem that causes the decision boundary of the attacked model to shift to the adversary's advantage. We evaluate our methodology against this poison attack and demonstrate its effectiveness. Other types of attacks focus on modifying uniquely the labels fed to the training model. Biggio et al. [5] study attacks where an adversary uniquely influences labels provided in the training process (a malicious annotator) and propose a kernel matrix correction defense for SVMs. Similarly, [12] present an attack that targets the labels input into the training system and a threshold based methodology to detect poison that relies on a Kappa statistic. Like RONI, this method requires that trusted, unpoisoned data is available. Finally, none of these approaches take into consideration provenance information associated with data points and labels during the training process to detect poison attacks.

7 CONCLUSION

In this paper, we present a novel methodology for detecting and filtering poisonous data collected to train an arbitrary supervised learning model. To the best of our knowledge, this is the first defense strategy that makes use of data provenance to prevent poisoning attacks. Trusted provenance information is available in many application scenarios such as in environmental sensing or even some social media environments. We present two variations of the provenance defense for both partially trusted and fully untrusted data sets. We evaluated our partially trusted approach using two previously proposed poison data generation methods. Our experimental results show that the detection effectiveness of the proposed provenance defense surpasses that of the baseline, thereby enabling the use of online and regularly re-trained machine learning models in adversarial environments where reliable provenance data can be obtained.

There are multiple interesting research avenues of future work. We are currently assuming that data sources are independent. As future work, it would be interesting to study cases where multiple data sources may collude to poison the machine learning model. Another promising avenue consists in investigating multiple calibration methodologies to detect how different provenance features may influence a change in accuracy of a particular model. Finally, we also plan to evaluate our fully untrusted model in more detail.

REFERENCES

[1] 2017. Data Provenance Model for Internet of Things (IoT) Systems. In *Service-Oriented Computing – ICSOC 2016 Workshops*. Springer Berlin Heidelberg, Berlin, Heidelberg.

[2] Muhammad Naveed Aman, Kee Chaing Chua, and Biplab Sikdar. 2017. Secure Data Provenance for the Internet of Things. In *Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security (IoTPTS '17)*. ACM, New York, NY, USA, 11–14. https://doi.org/10.1145/3055245.3055255

[3] Marco Barreno, Blaine Nelson, Anthony D Joseph, and JD Tygar. 2010. The security of machine learning. *Machine Learning* 81, 2 (2010), 121–148.

[4] Battista Biggio, Giorgio Fumera, and Fabio Roli. 2014. Security evaluation of pattern classifiers under attack. *IEEE transactions on knowledge and data engineering* 26, 4 (2014), 984–996.

[5] Battista Biggio, Blaine Nelson, and Pavel Laskov. 2011. Support vector machines under adversarial label noise. In *Asian Conference on Machine Learning*. 97–112.

[6] Battista Biggio, Blaine Nelson, and Pavel Laskov. 2012. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389* (2012).

[7] Aleksandar Chakarov, Aditya Nori, Sriram Rajamani, Shayak Sen, and Deepak Vijaykeerthy. 2016. Debugging machine learning tasks. *arXiv preprint arXiv:1603.07292* (2016).

[8] Jr Gadelha et al. Kairos: an architecture for securing authorship and temporal information of provenance data in grid-enabled workflow management systems. In *eScience'08*.

[9] Joseph Gardiner and Shishir Nagaraja. 2016. On the Security of Machine Learning in Malware C8C Detection: A Survey. *ACM Computing Surveys (CSUR)* 49, 3 (2016), 59.

[10] Ragib Hasan, Radu Sion, and Marianne Winslett. 2009. The Case of the Fake Picasso: Preventing History Forgery with Secure Provenance.. In *FAST*, Vol. 9. 1–14.

[11] John Lyle and Andrew Martin. 2010. Trusted computing and provenance: better together. In *Proceedings of the 2nd Workshop on the Theory and Practice of Provenance*. Usenix.

[12] Mehran Mozaffari-Kermani, Susmita Sur-Kolay, Anand Raghunathan, and Niraj K Jha. 2015. Systematic poisoning attacks on and defenses for machine learning in healthcare. *IEEE journal of biomedical and health informatics* 19, 6 (2015), 1893–1905.

[13] Blaine Nelson, Marco Barreno, Fuching Jack Chi, Anthony D Joseph, Benjamin IP Rubinstein, Udam Saini, Charles Sutton, JD Tygar, and Kai Xia. 2009. Misleading learners: Co-opting your spam filter. In *Machine learning in cyber trust*. Springer, 17–51.

[14] Blaine Nelson, Marco Barreno, Fuching Jack Chi, Anthony D Joseph, Benjamin IP Rubinstein, Udam Saini, Charles A Sutton, J Doug Tygar, and Kai Xia. 2008. Exploiting Machine Learning to Subvert Your Spam Filter. *LEET* 8 (2008), 1–9.

[15] Blaine Alan Nelson. 2010. *Behavior of Machine Learning Algorithms in Adversarial Environments*. University of California, Berkeley.

[16] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael Wellman. 2016. Towards the science of security and privacy in machine learning. *arXiv preprint arXiv:1611.03814* (2016).

[17] Mohammed Rangwala, Zhengli Liang, Wei Peng, Xukai Zou, and Feng Li. 2014. A Mutual Agreement Signature Scheme for Secure Data Provenance. *environments* 13 (2014), 14.

[18] Xinlei Wang, Kai Zeng, Kannan Govindan, and Prasant Mohapatra. 2012. Chaining for securing data provenance in distributed information networks. In *MILCOM 2012*. 1–6.

[19] Yan Zhou, Murat Kantarcioglu, Bhavani Thuraisingham, and Bowei Xi. 2012. Adversarial support vector machine learning. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1059–1067.